FACULTY WORKING PAPER 91-0114

Learning in a Partially Hard-Wired Recurrent Network

C.-M. Kuan
Department of Economics
University of Illinois

K. Hornik
Department of Economics
Technische Universität Wien, Austria

Bureau of Economic and Business Research
College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

FACULTY WORKING PAPER NO. 91-0114

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

February 1991

Learning in a Partially Hard-Wired Recurrent Network

C.-M. Kuan
Department of Economics
University of Illinois at Urbana-Champaign

and

K. Hornik
Institut für Statistik und Wahrscheinlichkeitstheorie
Technische Universität Wien, Vienna, Austria

Abstract

In this paper we propose a partially hard-wired Elman network. A distinct feature of our approach is that only minor modifications of existing on-line and off-line learning algorithms are necessary in order to implement the proposed network. This allows researchers to adapt easily to trainable recurrent networks. Given this network architecture, we show that in a general dynamic environment the standard back-propagation estimates for the learnable connection weights can converge to a mean square error minimizer with probability one and are asymptotically normally distributed.

1 Introduction

Neural network models have been successfully applied in a wide variety of disciplines. Typically, applications of networks with at least partially modifiable interconnection strengths are based on the so-called multilayer *feedforward* architecture, in which all signals are transmitted in one direction without feedbacks. In a dynamic context, however, a feedforward network may have difficulties in representing certain sequential behavior when its inputs are not sufficient to characterize temporal features of target sequences (Jordon, 1985). From the cognitive point of view, a feedforward network can perform only passive cognition, in that its outputs cannot be adjusted by an internal mechanism when static inputs are present (Norrod, O'Neill, & Gat, 1987). These deficiencies thus restrict the applicability of feedforward neural network models in dynamic environments.

In view of these problems, researchers have recently been studying *recurrent* networks, i.e., networks with feedback connections, see e.g., Jordon (1986), Elman (1988), Williams & Zipser (1988), and Kuan (1989). In a recurrent network, recurrent variables compactly summarize the past information and, together with other input variables, jointly determine the network outputs. Because recurrent variables are generated by the network, they are functions of the network connection weights. Owing to this parameter dependence, the standard back-propagation (BP) algorithm for feedforward networks cannot be applied because it fails to take the correct gradient search direction (cf. Rumelhart, Hinton, & Williams, 1986). Kuan, Hornik & White (1990) propose a recurrent BP algorithm generalizing the standard BP algorithm to various recurrent networks. However, this algorithm has quite complex updating equations and restrictions, and therefore cannot be used straightforwardly by recurrent networks practitioners.

In this paper we suggest an easier way to implement recurrent networks. We focus on a variant of the Elman (1988) network, in which only a subset of hidden unit activations serve as recurrent variables. We propose to hard-wire the connections between the recurrent units and their inputs. This approach has the following advantages. First, the resulting network avoids the aforementioned problem of parameter dependence. Second, the necessary constraints on recurrent connections suggested by Kuan, Hornik, & White (1990) can easily be imposed by hard-wiring. Third, off-line learning is made possible for the proposed network. Consequently, only minor modifications of existing on-line and off-line learning algorithms are needed. This is very convenient for neural network practitioners. Given this hard-wired network, we show that in general dynamic environments the resulting BP estimates converge to a mean squared error minimizer with probability one and are asymptotically normally distributed. Our convergence results extend the results of Kuan, Hornik, & White (1990) for general recurrent networks and are analogous to the results of Kuan & White (1990) for feedforward networks.

This paper proceeds as follows. In section 2 we briefly review recurrent networks. In section 3 we discuss a variant of the Elman network and its learning algorithms. We establish strong consistency and asymptotic normality of the learning estimates in section 4. Section 5 concludes the paper. Proofs are deferred to the appendix.

2 Recurrent Networks

A three layer recurrent network with k input units, l hidden units with common activation function ψ, and m output units with common activation function ϕ can

be written in the following generic form:

$$\begin{aligned} o_t &= \Phi(W a_t + v) \\ a_t &= \Psi(C x_t + D r_t + b) \\ r_t &= G(x_{t-1}, r_{t-1}, \theta), \end{aligned}$$

where the subscript t indexes time, x is the $k \times 1$ vector of network inputs, a is the $l \times 1$ vector of hidden unit activations, o is the $m \times 1$ vector of network outputs, Φ and Ψ compactly denote the unitwise activation rules in the output respectively hidden layer, and r_t is the $n \times 1$ vector of recurrent variables which is computed through some generic function G from the previous input x_{t-1}, the previous recurrent variable r_{t-1}, and

$$\theta = [\mathrm{vec}(C)', \mathrm{vec}(D)', \mathrm{vec}(W)', b', v']',$$

the vector of all network connection weights. (In what follows, $'$ denotes transpose, the vec operator stacks the columns of a matrix one underneath the other, and $|v|$ is the euclidean length of a vector v.)

More compactly, the above network can be written as

$$o_t = \Phi(W\Psi(C x_t + D r_t + b) + v) \tag{1}$$

$$r_t = G(x_{t-1}, r_{t-1}, \theta). \tag{2}$$

That is, the network output is jointly determined by the external inputs x and the recurrent variables r. Clearly, different choices of G yield different recurrent networks. When $r_t = o_{t-1}$ (output feedback),

$$r_t = G(x_{t-1}, r_{t-1}, \theta) = \Phi(W\Psi(C x_{t-1} + D r_{t-1} + b) + v),$$

and we obtain the Jordon (1986) network. When $r_t = a_{t-1}$ (hidden unit activation

feedbacks),

$$r_t = G(x_{t-1}, r_{t-1}, \theta) = \Psi(Cx_{t-1} + Dr_{t-1} + b),$$

and we have the Elman (1988) network.

By recursive substitution, (2) becomes

$$r_t = G(x_{t-1}, r_{t-1}, \theta) = G(x_{t-1}, G(x_{t-2}, r_{t-2}, \theta), \theta) = \cdots =: \ell_t(x^{t-1}, \theta),$$

where $x^{t-1} = (x_{t-1}, x_{t-2}, \ldots, x_0)$ is the collection of past inputs. Hence, r_t is a complex nonlinear function of θ and the entire past of x_t. In contrast with external input x_t, we may interpret r_t as "internal" input, in the sense that it is generated by the network. Given a recurrent network, the standard BP algorithm for feedforward networks does not perform correct gradient search over the parameter space because it fails to take the dependence of r_t on the learnable network weights into account. Consequently, meaningful convergence cannot be guaranteed (Kuan, 1989).

Kuan, Hornik, & White (1990) propose a recurrent BP algorithm which, by carefully calculating the correct gradients and including additional derivative updating equations, maintains the desired gradient search property. To ensure proper convergence behavior, their results also suggest some restrictions on the network connection weights. That is, parameters estimates are projected into some "stability" region whenever they violate the imposed constraints. Thus, much more effort is needed in programming appropriate learning algorithms for recurrent networks. Moreover, some of their conditions to ensure convergence of the recurrent BP algorithm are rather stringent.

3 A Partially Hard-Wired Elman Network

In this section we suggest an easier way to implement a variant of the Elman (1988) network. As we have discussed in section 2, improper convergence of the learning algorithms is mainly due to the dependence of the internal inputs r_t on the modifiable network parameters. To circumvent this problem, we propose to modify the Elman network as is depicted in figure 1.

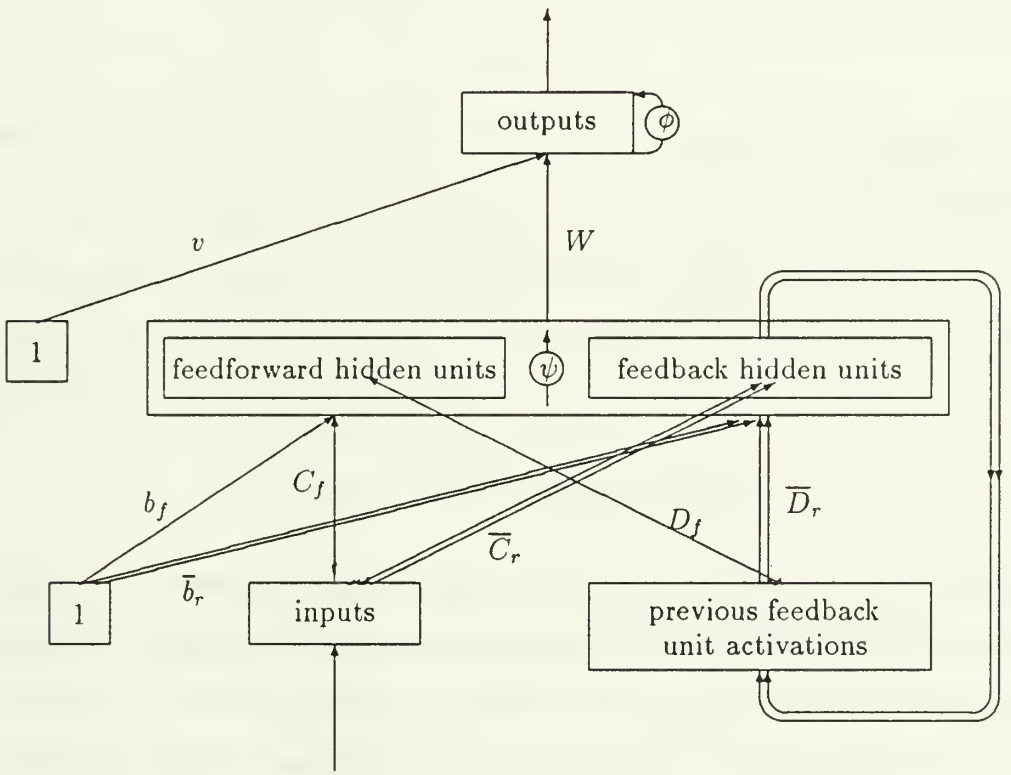

Figure 1. The proposed partially hard-wired recurrent network. Modifiable and hard-wired connections are represented by $\longrightarrow$ respectively $\Rightarrow$.

The hidden units are partitioned into two groups containing l_f respectively $l_r = l - l_f$ units, and only the units in the second group serve as recurrent units. Intuitively, the units in the first group play the standard role in artificial neural

networks, whereas the task of the recurrent units is to "index" information on previous inputs. Furthermore, the connections between the recurrent units and their inputs are hard-wired.

Hence, a is partitioned as $a = [a_f', a_r']'$, where a_f is the $l_f \times 1$ vector of activations of the (purely feedforward) hidden units in the first group, and a_r is the $l_r \times 1$ vector of activations of the feedback (recurrent) hidden units in the second group such that

$$r_t = a_{r,t-1}.$$

If the connection matrices C and D and the bias vector b are partitioned comformably as

$$C = \begin{bmatrix} C_f \\ \overline{C}_r \end{bmatrix}, \qquad D = \begin{bmatrix} D_f \\ \overline{D}_r \end{bmatrix}, \qquad b = \begin{bmatrix} b_f \\ \overline{b}_r \end{bmatrix},$$

then

$$\begin{aligned} a_{f,t} &= \Psi(C_f x_t + D_f a_{r,t-1} + b_f) \\ a_{r,t} &= \Psi(\overline{C}_r x_t + \overline{D}_r a_{r,t-1} + \overline{b}_r), \end{aligned}$$

where now $\overline{C}_r$, $\overline{D}_r$ and $\overline{b}_r$ are fixed due to hard-wiring. Different choices of $\overline{C}_r$, $\overline{D}_r$ and $\overline{b}_r$ determine how the past information should be represented, hence they are problem-dependent and should be left to researchers.

Hence, writing the proposed network in a nonlinear functional form, we have

$$o_t = \Phi(W\Psi(Cx_t + Da_{r,t-1} + b) + v) =: F(x_t, a_{r,t-1}, \theta) \tag{3}$$

and

$$r_t = a_{r,t-1} = \Psi(\overline{C}_r x_{t-1} + \overline{D}_r a_{r,t-2} + \overline{b}_r) =: G(x_{t-1}, a_{r,t-2}, \overline{\theta}), \tag{4}$$

where now

$$\theta = [\mathrm{vec}(W)', \mathrm{vec}(C_f)', \mathrm{vec}(D_f)', b_f', v']'$$

is the $p \times 1$ vector which contains all the learnable network weights, where $p := m(l+1) + l_f(k + l_r + 1)$, and

$$\overline{\theta} = [\text{vec}(\overline{C}_r)', \text{vec}(\overline{D}_r)', \bar{b}_r']'$$

contains all the hard-wired weights. By recursive substitution, (4) becomes

$$r_t = G(x_{t-1}, r_{t-1}, \overline{\theta}) = G(x_{t-1}, G(x_{t-2}, r_{t-2}, \overline{\theta}), \overline{\theta}) = \cdots =: \mu_t(x^{t-1}, \overline{\theta}),$$

cf. equation (2). Thus, $r_t = a_{r,t-1}$ is a function of the entire past of x_t and the hard-wired weights $\overline{\theta}$.

Because r_t is not a function of the learnable weights θ, the aforementioned problem of parameter dependence is thus avoided. It follows that the standard BP algorithm for feedforward networks is applicable to the proposed network with respect to the learnable weigths θ. Letting y_t denote the target pattern presented at time t, the BP algorithm is

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \eta_t \nabla_\theta F(x_t, r_t, \hat{\theta}_t)(y_t - F(x_t, r_t, \hat{\theta}_t)), \tag{5}$$

where η_t is learning rate employed at time t and $\nabla_\theta F$ is the matrix of partial derivatives of F with respect to the components of θ. However, in both theory and practice it is necessary to keep the BP estimates in some compact subset Θ of $\mathbb{R}^p$, thus preventing the entries from becoming extremely large. This, being a typical requirement in the convergence analysis of the BP type of algorithms, see e.g., Kuan & White (1990) and Kuan, Hornik, & White (1990), can, if not automatically guaranteed by the algorithm, be accomplished by applying a projection operator π which maps $\mathbb{R}^p$ onto Θ to the BP estimates. Usually, a truncation device is convenient for this purpose. This requirement entails little loss because it is usually inactive when very large trunaction bounds are imposed.

In light of (5), we only have to modify the existing BP algorithm slightly to incorporate the internal inputs a_r into the algorithm. Furthermore, if a fixed training data set is given, the internal inputs $a_{r,t}$ can be calculated first, and off-line learning methods such as nonlinear least squares can then be applied to estimate the learnable weights θ. These advantages allow researchers to adapt to recurrent networks quite easily. It is then interesting to know the properties of the algorithm (5) applied to the proposed network given by (3) and (4). This is the topic to which we now turn.

4 Asymptotic Properties of the BP Algorithm

Let $\{V_t\}$ be some sequence of random variables defined on a probability space $(\Omega, \mathcal{F}, P)$, $\mathcal{F}_\tau^t$ be the σ-algebra generated by $V_\tau, V_{\tau+1}, \ldots, V_t$, and let $\{Z_t\}$ be a sequence of square integrable random variables on that probability space. We write $E_{t-m}^{t+m}(Z_t)$ for the conditional expectation $E(Z_t|\mathcal{F}_{t-m}^{t+m})$ and $\|\cdot\|$ for the norm in $L_2(P)$, i.e., $\|Z\| = (E|Z|^2)^{1/2}$.

Definition 4.1. Let

$$\nu_m := \sup_t \|Z_t - E_{t-m}^{t+m}(Z_t)\|.$$

Then $\{Z_t\}$ is *near epoch dependent* (NED) *on* $\{V_t\}$ *of size* $-a$ if for some $\lambda < -a$, $\nu_m = O(m^\lambda)$ as $m \to \infty$.

This definition conveys the idea that a random variable depends essentially on the information generated by "more or less current" V_t and does not depend too much on the information contained in the distant future or past. The larger the magnitude of the size of ν_m, the faster the dependence of the remote information dies out. More details on near epoch dependence can be found in Billingsley (1968), McLeish (1975), and Gallant & White (1988).

The lemma below ensures that recurrent variables are well behaved and do not have too long memory.

Lemma 4.2. *Let $\{r_t\}$ be generated by* (4), *where $\{x_t\}$ is NED on $\{V_t\}$ of size $-a$ and the common hidden unit activation function ψ is bounded and continuously differentiable with bounded first derivative. If $|\mathrm{vec}(\overline{D}_r)| < M_\psi^{-1}$, where $M_\psi := \sup_{\sigma \in R} |\psi'(\sigma)|$, then $\{r_t\}$ is a bounded sequence NED on $\{V_t\}$ of size $-a$.*

Remark 1. Notice that if the input data $\{x_t\}$ form a sequence of independent random variables (which is a special case of an NED sequence), then $\{r_t\}$ need not necessarily be mixing but is NED on $\{x_t\}$ of arbitrarily large size, see Gallant & White (1988, pp. 27-31). Hence, introducing the concept of near epoch dependence is not a technical triviality, but a necessity when dealing with feedback networks in stochastic input environments.

In what follows we compactly write the algorithm (5) as

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \eta_t \, h_t(\hat{\theta}_t),$$

where $z_t = (y_t', x_t')'$ and $h_t(\theta) = \nabla_\theta F(x_t, r_t, \theta)(y_t - F(x_t, r_t, \theta))$. Our consistency result is based on the ordinary differential equation (ODE) method of Kushner & Clark (1978), cf. Ljung (1977). This approach is now well-known in analyzing neural network learning algorithms, cf. e.g., Oja (1982), Oja & Karhunen (1985), Sanger (1989), Kuan & White (1990), Kuan, Hornik, & White (1990), and Hornik & Kuan (1990).

We need the following notation. Let $\tau_0 = 0$ and, for $t \geq 1$, let $\tau_t := \sum_{i=0}^{t-1} \eta_i$. The piecewise linear interpolation of $\{\hat{\theta}_t\}$ with interpolation intervals $\{\eta_t\}$ is

$$\theta^0(\tau) = \left(\frac{\tau_{t+1} - \tau}{\eta_t}\right) \hat{\theta}_t + \left(\frac{\tau - \tau_t}{\eta_t}\right) \hat{\theta}_{t+1}, \qquad \tau \in [\tau_t, \tau_{t+1}),$$

and for each t, its "left shift" is

$$\theta^t(\tau) = \theta^0(\tau_t + \tau).$$

Observe in particular that $\theta^t(0) = \theta^0(\tau_t) = \hat{\theta}_t$.

We impose the following conditions.

A.1. $\{V_t\}$ and $\{z_t\}$ are defined on a complete probability space $(\Omega, \mathcal{F}, P)$ such that for some $r \geq 4$,

(i) $\{V_t\}$ is a mixing sequence with mixing coefficients ϕ_m of size $-r/2(r-1)$ or α_m of size $-r/(r-2)$ and

(ii) the sequence $\{z_t\}$ is NED on $\{V_t\}$ of size -1 with $\sup_t |x_t| \leq M_x < \infty$ and $\sup_t E(|y_t|^r) < \infty$.

A.2. For the network architecture as specified in (3) and (4),

(i) ϕ and ψ are continuously differentiable of order 3. ψ is bounded and has bounded first order derivative.

(ii) $|\text{vec}(\overline{D}_r)| < M_\psi^{-1}$, where $M_\psi = \sup_{\sigma \in \mathbb{R}} |\psi'(\sigma)|$.

A.3. $\{\eta_t\}$ is a sequence of positive real numbers such that $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$.

A.4. For each $\theta \in \Theta$, $\overline{h}(\theta) = \lim_t E(h_t(\theta))$ exists.

A.1 allows the data to exhibit a considerable amount of dependence in the sense that they are functions of the (possibly infinite) history of an underlying mixing sequence. For more details on α- and ϕ-mixing sequences we refer to White (1984). Assuming that the external inputs x_t are uniformly bounded simplifies some technicalities needed to establish convergence and causes no loss of generality, as

pointed out by Kuan & White (1990). Desired generality is assured by allowing the y_t sequence to be unbounded. Note that typical choices for ψ such as the logistic squasher and hyperbolic tangent squasher satisfy A.2(i). Condition A.2(ii) is needed in lemma 4.2 and is the constraint suggested by Kuan, Hornik & White (1990) for general recurrent networks. A.3 is a typical restriction on the learning rates for BP types of algorithms. For example, learning rates of order $1/t$ satisfy this condition. A.4 is needed to define the associated ODE whose solution trajectory is the limiting path of the interpolated processes $\{\theta^t(\cdot)\}$.

The result below follows from corollary 3.5 of Kuan & White (1990).

Theorem 4.3. *For the network given by* (3) *and* (4) *and the algorithm* (5), *suppose that assumptions* A.1-A.4 *hold. Then*

(a) $\{\theta^t(\cdot)\}$ *is bounded and equicontinuous on bounded intervals with probability one, and all limits of convergent subsequences satisfy the ODE* $\dot{\theta} = \overline{h}(\theta)$.

(b) *Let* Θ^* *be the set of all (locally) asymptotically stable equilibria of this ODE contained in* Θ, *and let* $\mathcal{D}(\Theta^*) \subset \mathbb{R}^p$ *be the domain of attraction of* Θ^*. *Then, if* $\hat{\theta}_t$ *enters a compact subset of* $\mathcal{D}(\Theta^*)$ *infinitely often with probability one, and thus in particular, if* $\Theta \subseteq \mathcal{D}(\Theta^*)$, *then with probability one,* $\theta_t \to \Theta^*$ *as* $t \to \infty$.

Remark 2. Because the elements θ^* of Θ^* solve the equation $\lim_t E(h(z_t, r_t, \theta)) = \overline{h}(\theta) = 0$, they (locally) minimize

$$\lim_t E|y_t - F(x_t, r_t, \theta)|^2. \tag{6}$$

Theorem 4.3 thus shows that the BP estimates can converge to a mean squared error minimizer with probability one. Note however that this convergence occurs conditional on $\overline{\theta}$.

Remark 3. By the Toeplitz lemma, $\lim_T T^{-1} \sum_{t=1}^{T} E|y_t - F(x_t, r_t, \theta)|^2$ is the same as (6). Therefore, the (on-line) BP estimates converge to the same limit as the (off-line) nonlinear least squares estimator.

Remark 4. As y_t is not required to be bounded, our strong consistency result holds under less stringent conditions than those of Kuan, Hornik & White (1990) for the fully recurrent BP algorithm.

To establish asymptotic normality we consider the algorithm (5) with the specific choice $\eta_t = (t+1)^{-1}$. (Note that no limiting distribution results for BP estimators in recurrent networks have been published thus far; in particular, Kuan, Hornik, & White (1990) give only a consistency result for their recurrent BP algorithm.) Let $U_t := (t+1)^{1/2}(\hat{\theta}_t - \theta^*)$ be the sequence of normalized estimates. The piecewise constant interpolation of U_t on $[0, \infty)$ with interpolation intervals $\{(t+1)^{-1}\}$ is defined as

$$\overline{U}(\tau) = U_t, \qquad \tau \in [\tau_t, \tau_{t+1}),$$

and again, for each t its "left shift" is defined as

$$U^t(\tau) = \overline{U}(\tau_t + \tau), \qquad \tau \geq 0.$$

Finally, let

$$\overline{H}(\theta) := \lim_t E[\nabla_\theta h_t(\theta)] + I_p/2,$$

where I_p is the p-dimensional identity matrix.

Our result follows from the stochastic differential equation (SDE) approach of Kushner & Huang (1979). In contrast with the ODE approach, the interpolated processes can now shown to converge *weakly* to the solution paths of a corresponding SDE with respect to the Skorohod topology. For more details on weak

convergence we refer to Billingsley (1968). The following conditions suffice for the asymptotic normality result.

B.1. A.1(i) holds, and $\{z_t\}$ is a stationary sequence NED on $\{V_t\}$ of size -8 with $\sup_t |x_t| \leq M_x < \infty$ $\sup_t E(|y_t|^8) < \infty$.

B.2. A.2 holds with ϕ and ψ continously differentiable of order 4.

B.3. $\theta^* \in \text{int}(\Theta)$ is such that $\overline{h}(\theta^*) = 0$ and all eigenvalues of $\overline{H}(\theta^*)$ have negative real parts.

The result below follows from corollary 3.6 of Kuan & White (1990).

Theorem 4.4. *Consider the network given by* (3) *and* (4) *and the algorithm* (5) *with* $\eta_t = (t+1)^{-1}$, *suppose that assumptions* B.1-B.3 *hold and that with probability one,* $\theta_t \to \theta^*$ *as* $t \to \infty$. *Then* $\{U^t(\cdot)\}$ *converges weakly to the stationary solution of the stochastic differential equation*

$$dU(\tau) = \overline{H}(\theta^*)U(\tau)\,d\tau + \overline{\Sigma}(\theta^*)^{1/2}\,dW(\tau),$$

where W *denotes the standard p-variate Wiener process and*

$$\overline{\Sigma}(\theta^*) := \lim_t \sum_{j=-\infty}^{\infty} E[h_t(\theta^*)h_{t+j}(\theta^*)'].$$

In particular,

$$(t+1)^{1/2}(\hat{\theta}_t - \theta^*) \xrightarrow{\mathcal{D}} N(0, S(\theta^*)),$$

where "$\xrightarrow{\mathcal{D}}$*" signifies convergence in distribution and*

$$S(\theta^*) := \int_0^{\infty} \exp(\overline{H}(\theta^*)s)\,\overline{\Sigma}\,\exp(\overline{H}(\theta^*)s)\,ds$$

is the unique solution to the matrix equation $\overline{H}(\theta^*)S + S\overline{H}(\theta^*)' = -\overline{\Sigma}(\theta^*)$

Remark 5. If $\eta_t = (t+1)^{-1}R$, where R is a nonsingular $p \times p$ matrix, the SDE in theorem 4.4 becomes $dU(\tau) = \overline{H}(\theta^*)U(\tau)\,d\tau + R\overline{\Sigma}(\theta^*)^{1/2}\,dW(\tau)$, and the covariance matrix of the asymptotitc distribution of $\hat{\theta}_t$ becomes $RS(\theta^*)R'$.

Remark 6. If the probability that $\hat{\theta}_t$ converges to θ^* is positive, but less than one, the above theorem provides the limiting distribution *conditional* on convergence to θ^*. Hence, if Θ^* contains only finitely many points, assumption B.3 is satisfied for each $\theta^* \in \Theta^*$, and $\hat{\theta}_t$ converges with probability one to one of the elements of Θ^*, then the asymptotic distribution of $\hat{\theta}_t$ is a mixture of $N(\theta^*, S(\theta^*))$ distributions, weighted relative to the convergence probabilities.

5 Conclusions

In this paper we propose a partially hard-wired Elman network, in which only a subset of hidden-unit activations is allowed to feed back into the network and connections between these hidden units and input layer are hard-wired. A distinct feature of our approach is that existing on-line and off-line learning algorithms can be slightly modified to implement the proposed network. (Note that off-line learning is not possible for a fully learnable recurrent network.) This is particularly convenient for researchers. Our results also show that the estimates from the standard BP algorithm adapted to this network can converge to a mean squared error minimizer with probability one and are asymptotically normally distributed. These asymptotic properties are analogous to those of the standard and recurrent BP algorithms.

As the convergence results in this paper are conditional on the hard-wired connection weights $\overline{\theta}$, the resulting weight estimates are not fully optimal, in contrast with fully learnable recurrent networks. To improve the performance of the

proposed network, one can train the network with various hard-wired connection weights and search for the best performing architecture.

Appendix

Lemma A. *Let $\{x_t\}$ be NED on $\{V_t\}$ of size $-a$ and let the square integrable sequence $\{r_t\}$ be generated by the recursion*

$$r_t = G(x_{t-1}, r_{t-1}, \overline{\theta}).$$

Suppose that $G(\cdot, r, \overline{\theta})$ satisfies a Lipschitz condition uniformly in r, i.e., there exists a finite constant L such that for all r,

$$|g(x_1, r, \overline{\theta}) - g(x_2, r, \overline{\theta})| \leq L|x_1 - x_2|,$$

and that $G(x, \cdot, \overline{\theta})$ is a contraction mapping uniformly in x, i.e., there exists some $\rho < 1$ such that for all x,

$$|G(x, r_1, \overline{\theta}) - g(x, r_2, \overline{\theta})| \leq \rho|r_1 - r_2|.$$

Then $\{r_t\}$ is NED on $\{V_t\}$ of size $-a$.

Proof. We first observe that

$$\begin{aligned}
&\|r_t - E_{t-m}^{t+m}(r_t)\| \\
&\quad = \|G(x_{t-1}, r_{t-1}, \overline{\theta}) - E_{t-m}^{t+m}(G(x_{t-1}, r_{t-1}, \overline{\theta}))\| \\
&\quad \leq \|G(x_{t-1}, r_{t-1}, \overline{\theta}) - G(E_{t-m}^{t+m-2}(x_{t-1}), E_{t-m}^{t+m-2}(r_{t-1}), \overline{\theta})\| \\
&\quad \leq \|G(x_{t-1}, r_{t-1}, \overline{\theta}) - G(E_{t-m}^{t+m-2}(x_{t-1}), r_{t-1}, \overline{\theta})\| \\
&\qquad\qquad + \|G(E_{t-m}^{t+m-2}(x_{t-1}), r_{t-1}, \overline{\theta}) - G(E_{t-m}^{t+m-2}(x_{t-1}), E_{t-m}^{t+m-2}(r_{t-1}), \overline{\theta})\| \\
&\quad \leq L\|x_{t-1} - E_{t-m}^{t+m-2}(x_{t-1})\| + \rho\|r_{t-1} - E_{t-1}^{t+m-2}(r_{t-1})\|,
\end{aligned}$$

where the first inequality follows from the fact that $E_{t-m}^{t+m}(G(x_{t-1}, r_{t-1}, \overline{\theta}))$ is the best mean square predictor of $G(x_{t-1}, r_{t-1}, \overline{\theta})$ among all $\mathcal{F}_{t-m}^{t+m}$-measurable functions and the second inequality follows from the triangle inequality. Hence, we

obtain

$$\nu_{r,m} \leq L\nu_{x,m-1} + \rho\nu_{r,m-1}, \tag{a1}$$

where $\nu_{x,m}$ and $\nu_{r,m}$ are the NED coefficients for $\{x_t\}$ and $\{r_t\}$, respectively. We must show that for some $\lambda < -a$, $\nu_{r,m}$ is $O(m^\lambda)$ as $m \to \infty$. Because $\{x_t\}$ is NED on $\{V_t\}$ of size a, we can find a finite constant C_0 and some $\lambda_0 < -a$ such that $\nu_{x,m} \leq C_0 m^{\lambda_0}$. By the fact that $\rho < 1$, we can find m_0 and some $\sigma > 1$ such that $\rho\sigma < 1$ and for all $m \geq m_0$,

$$(m/(m+1))^{\lambda_0} \leq \sigma.$$

Let

$$D_0 := \max\left\{\frac{\nu_{r,m_0}}{m_0^{\lambda_0}}, \frac{C_0 L\sigma}{1-\rho\sigma}\right\}.$$

We now prove by induction that for all $m \geq m_0$, $\nu_{r,m} \leq D_0 m^{\lambda_0}$. For $m = m_0$, this is trivially true by the definition of D_0. Suppose we have already shown that for some $m \geq m_0$, $\nu_{r,m} \leq D_0 m^{\lambda_0}$. Then, using (a1),

$$\begin{aligned}
\nu_{r,m+1} &\leq L\nu_{x,m} + \rho\nu_{r,m} \\
&\leq LC_0 m^{\lambda_0} + \rho D_0 m^{\lambda_0} \\
&= (LC_0 + \rho D_0)(m+1)^{\lambda_0}(m/(m+1))^{\lambda_0} \\
&\leq (LC_0 + \rho D_0)\sigma(m+1)^{\lambda_0} \\
&\leq D_0(m+1)^{\lambda_0},
\end{aligned}$$

completing the induction step and thus the proof of the lemma.

Proof of Lemma 4.2. By boundedness of ψ, the sequence $\{r_t\}$ generated by (4) is bounded and thus trivially square integrable. Hence, in view of the above lemma A, it suffices to show that G is Lipschitz continuous in x and a contraction mapping in r. As by assumption the first derivative of ψ is uniformly bounded, G

is clearly Lipschitz continuous in x with Lipschitz constant $L = M_\psi|\overline{C}_r|$. (If A is a matrix, then $|A| := \max\{|Ax| : |x| = 1\}$.) Similarly, let $\nabla_r G$ denote the matrix of partial derivatives of G with respect to r. Note that $|\nabla_r G(x, r, \overline{\theta})|$ is the square root of the maximal singular value of $\nabla_r G$, and thus by a well-known result from linear algebra,

$$\begin{aligned} |\nabla_r G(x, r, \overline{\theta})| &\leq \left(\text{trace}\,(\nabla_r G(x, r, \overline{\theta}) \nabla_r G(x, r, \overline{\theta})')\right)^{1/2} \\ &\leq M_\psi (\text{trace}\,(\overline{D}_r \overline{D}_r')^{1/2} \\ &= M_\psi |\text{vec}(\overline{D}_r)| \\ &=: \rho. \end{aligned}$$

By assumption, $\rho < 1$. As clearly,

$$|G(x, r_1, \overline{\theta}) - G(x, r_2, \overline{\theta})| \leq \sup_r |\nabla_r G(x, r, \overline{\theta})|\ |r_1 - r_2| \leq \rho |r_1 - r_2|,$$

G is a contraction mapping in r, thereby completing the proof of lemma 4.2.

Proof of theorem 4.3. We verify the conditions of corollary 3.5 of Kuan & White (1990), which we shall briefly refer to as [KW]. Their conditions A.4 and C.3 are explicitly assumed (our assumptions A.3 and A.4). It follows from lemma 4.2 that $\{r_t\}$ and thus also $\{\xi_t\}$ are bounded sequences NED on $\{V_t\}$ of size -1, where $\xi_t = [x_t', r_t']'$, which establishes condition C.1 of [KW]. Let M_ξ be an upper bound for the sequence $\{\xi_t\}$, and let $K_\xi := \{\xi : |\xi| \leq M_\xi\}$. Condition C.2 of [KW] requires that in $K_\xi \times \Theta$, both $F(\xi, \cdot)$ and $\nabla_\theta F(\xi, \cdot)$ satisfy a Lipschitz condition with Lipschitz constants $L_1(\xi)$ and $L_2(\xi)$, respectively, where L_1 and L_2 are Lipschitz continuous in ξ, and that both $F(\cdot, \theta)$ and $\nabla_\theta F(\cdot, \theta)$ satisfy a Lipschitz condition. It is straightforward to show that continuous differentiability of A.2(i) ensures these Lipschitz conditions. See also corollary 4.1 of Kuan & White (1990).

Proof of theorem 4.4. We verify the conditions of corollary 3.6 of [KW]. Lemma 4.2 ensures that $\{r_t\}$ is NED on $\{V_t\}$ of size -8. Stationarity of $\{x_t\}$ implies that $\{r_t\}$ is also stationary. Hence, $\{\xi_t\}$ is a stationary sequence NED on $\{V_t\}$ of size -8, which estabishes condition D.1 of [KW]. Condition D.2 of [KW] follows from B.3 and the moment condition of B.1. Finally, as in the preceding proof, four times continuous differentiability of B.2 ensures the Lipschitz conditions imposed in condition D.3 of [KW]. See also corollary 4.2 of Kuan & White (1990).

References

Billingsley, P. (1968). *Convergence of probability measures.* New York: Wiley.

Elman, J. L. (1988). *Finding structure in time.* CLR Report 8801, Center for Research in Language, University of California, San Diego.

Gallant, A. R. & White, H. (1988). *A unified theory of estimation and inference for nonlinear dynamic models.* Oxford: Basil Blackwell.

Hornik, K., & Kuan, C.-M. (1990). *Convergence analysis of local feature extraction algorithms.* BEBR Discussion Paper 90-1717, College of Commerce, University of Illinois, Urbana-Champaign.

Jordon, M. I. (1985). *The learning of representations for sequential performance.* Ph.D. Dissertation, University of California, San Diego.

Jordon, M. I. (1986). *Serial order: a parallel distributed processing approach.* ICS Report 8604, Institute for Cognitive Science, University of California, San Diego.

Kuan, C.-M. (1989). *Estimation of neural network models.* Ph.D. thesis, Department of Economics, University of California, San Diego.

Kuan, C.-M., Hornik, K., & White, H. (1990). Some convergence results for learning in recurrent neural networks. *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems*, Ed. K. S. Narendra, New Haven: Yale University, 103–109.

Kuan, C.-M., & White, H. (1990). *Recursive M-estimation, nonlinear regression and neural network learning with dependent observations.* BEBR Working Paper 90-1703, College of Commerce, University of Illinois, Urbana-Champaign.

Kushner, H. J., & Clark, D. S. (1978). *Stochastic approximation methods for constrained and unconstrained systems.* New York: Springer Verlag.

Kushner, H. J., & Huang, H. (1979). Rates of convergence for stochastic approximation type algorithms. *SIAM Journal of Control and Optimization*, **17**, 607–617.

Ljung, L. (1977). Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, **AC-22**, 551–575.

McLeish, D. (1975). A maximal inequality and dependent strong laws. *Annals of Probability*, **3**, 829–839.

Norrod, F. E., O'Neill, M. D., & Gat, E. (1987). Feedback-induced sequentiality in neural networks. In *Proceedings of the IEEE First International Conference on Neural Networks* (pp. II: 251–258). San Diego: SOS Printing.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematics and Biology*, **15**, 267–273.

Oja, E., & Karhunen, J. (1985). On stochastic approximation of the eigenvectors and the eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, **106**, 69–84.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & The PDP Research Group, *Parallel distributed processing: Explorations in the microstructures of cognition*, (pp. I: 318–362). Cambridge, MA: MIT Press.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, **2**, 459–473.

Williams, R. J., & Zipser, D. (1988). *A learning algorithm for continually running fully recurrent neural networks.* ICS Report 8805, Institute of Cognitive Science, University of California, San Diego.